LINKgroup

Distance Learning System

pandas

# Pandas biblioteka

Python Data Access

# Šta je pandas


- Pandas je Python biblioteka za analizu podataka
- Ima mogućnost tabelarne manipulacije podacima
- U osnovi ima NumPy biblioteku
- Pandas takođe ima integrisanu pyplot biblioteku

*pip install pandas*

# Pandas strukture

- Pandas sadrži dve bitne strukture: **Series** i **DataFrame**

```
import matplotlib.pyplot as plt
import pandas as pd
```

- Series je niz vrednosti

```
dt1 = pd.Series([1,2,3,4])
dt2 = pd.Series([2,3,4,5])
```

```
0    1
1    2
2    3
3    4
```

- DataFrame je niz Series objekata

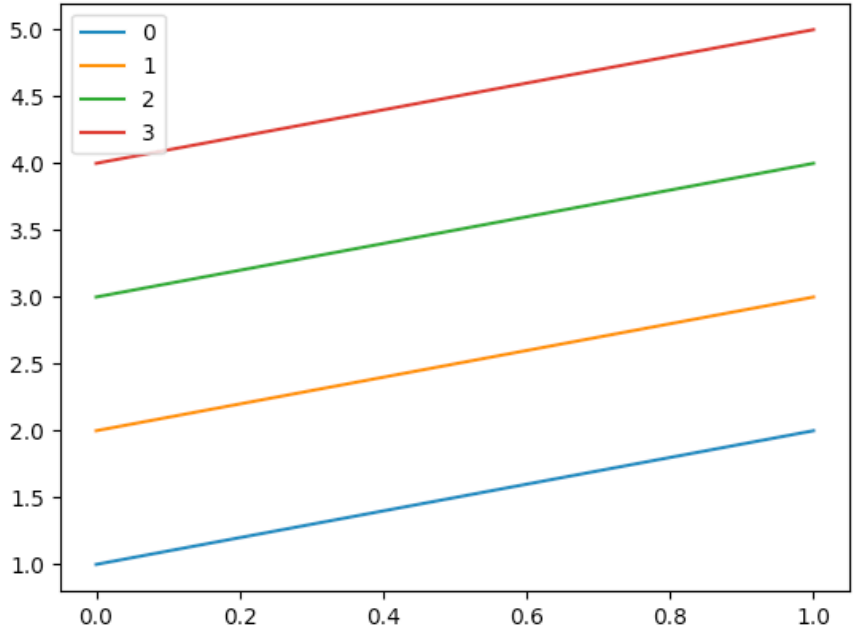
```
dt = pd.DataFrame([dt1,dt2])
```

```
   0  1  2  3
0  1  2  3  4
1  2  3  4  5
```

# Iscrtavanje rezultata

- Pandas ima podršku za pyplot biblioteku

```
dt1 = pd.Series([1,2,3,4])
dt2 = pd.Series([2,3,4,5])
dt = pd.DataFrame([dt1,dt2])
dt.plot()
plt.show()
```



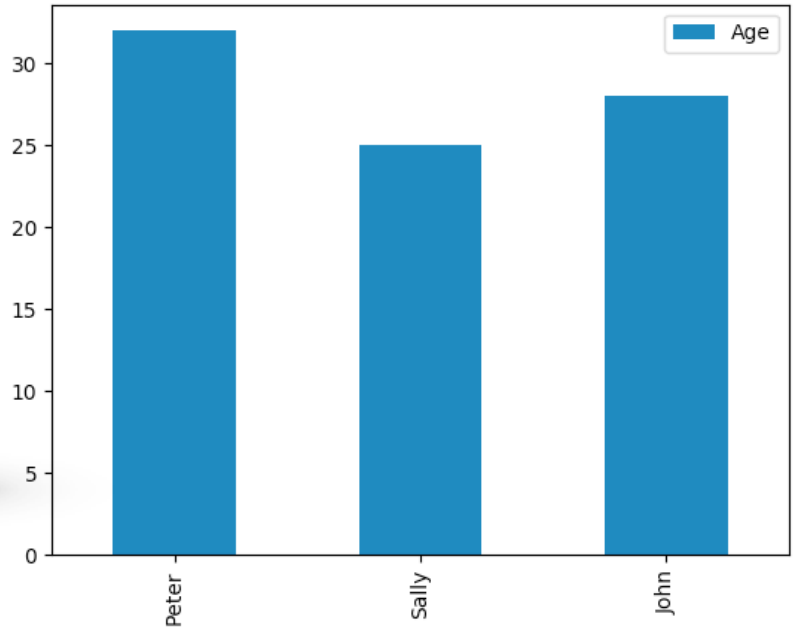
# Definisanje kolona

```
dt1 = pd.Series([1,2,3,4],index=['a','b','c','d'])
dt2 = pd.Series([2,3,4,5],index=['a','b','c','d'])
dt = pd.DataFrame([dt1,dt2])
```

| | a | b | c | d |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 |

# Definisanje kolona

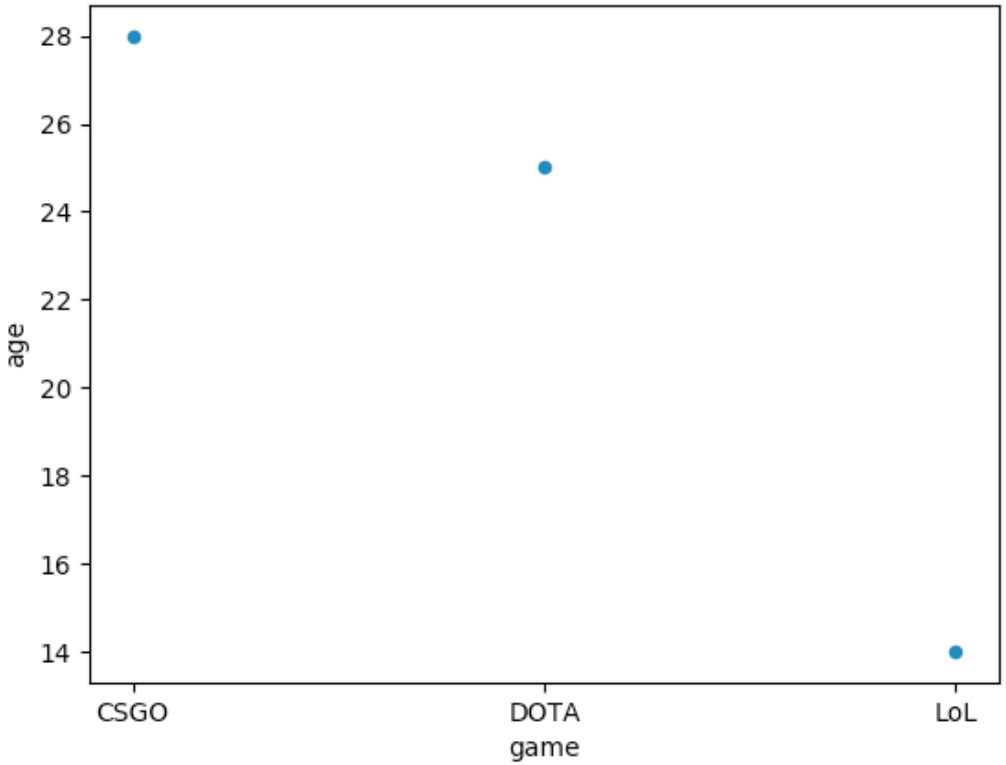
```
dt = pd.DataFrame(
    [["Peter",32],["Sally",25],["John",28]],
    columns=["User","Age"]
)
dt.plot.bar(x="User",y="Age")
plt.show()
```



# Definisanje kolona

```
dt = pd.DataFrame(
    [
        ["Peter","CSGO",28],
        ["Peter","DOTA",25],
        ["Sally","LoL",14]],
        columns=["user","game","age"]
)
dt.plot.scatter(x="game",y="age")
plt.show()
```

| | user | game | age |
|---|---|---|---|
| 0 | Peter | CSGO | 28 |
| 1 | Peter | DOTA | 25 |
| 2 | Sally | LoL | 14 |



# Učitavanje i čuvanje podataka

https://pandas.pydata.org/docs/user_guide/io.html

- Pandas može da učita i parsira različite formate
- Čitanje se obavlja **read** serijom funkcija nad pandas objektom

```
tbl = pd.read_csv("myfile.csv")
```

- Upise se obavlja **to** serijom funkcija, nad nekom pandas strukturom

```
tbl = pd.DataFrame(
    [["Peter","CSGO",25],["Sally","DOTA",32],["John","LoL",19]],
    columns=["user","game","age"])
tbl.to_csv('myfile.csv')
```

# Pregled podataka

```
print(dt.describe())
```

```
              age
count    3.000000
mean    22.333333
std      7.371115
min     14.000000
25%     19.500000
50%     25.000000
75%     26.500000
max     28.000000
```

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   user    3 non-null      object
 1   game    3 non-null      object
 2   age     3 non-null      int64
dtypes: int64(1), object(2)
memory usage: 200.0+ bytes
```

# Filtracija podataka

```
print(dt[dt["age"]>14])
```



| | user | game | age |
|---|---|---|---|
| 0 | Peter | CSGO | 28 |
| 1 | Peter | DOTA | 25 |

```
print(dt.head(2))
```

| | user | game | age |
|---|---|---|---|
| 0 | Peter | CSGO | 28 |
| 1 | Peter | DOTA | 25 |

```
print(dt.tail(2))
```

| | user | game | age |
|---|---|---|---|
| 1 | Peter | DOTA | 25 |
| 2 | Sally | LoL | 14 |

# Sortiranje podataka

```
tbl = tbl.groupby("player_id").max()
tbl = tbl.sort_values('game', ascending=False)
print(tbl.head(10))
```

| player_id | game |
|---|---|
| 1621779 | 51 |
| 1501928 | 51 |
| 1605388 | 51 |
| 673993 | 51 |
| 1712956 | 51 |
| 1399136 | 51 |
| 669015 | 51 |
| 1713365 | 51 |
| 1713837 | 51 |
| 660952 | 51 |

# Agregacija i grupisanje

```
print(dt.groupby("user").count())
```



| user | game | age |
|---|---|---|
| Peter | 2 | 2 |
| Sally | 1 | 1 |

# Pivoting

```
dt = pd.DataFrame(
    [
        ["Peter","CSGO",25],
        ["Sally","DOTA",32],
        ["John","LoL",18],
        ["Sam","DOTA",21]
    ],
    columns=["user","game","age"]
)
dt = pd.pivot_table(dt,
    values=["age"],
    columns="game",
    aggfunc=numpy.min
)
```

| game | CSGO | DOTA | LoL |
|---|---|---|---|
| age | 25 | 21 | 18 |

**kolone za agregaciju**

**kolone za prikaz**

**agregatna funkcija**

# Rad sa vremenom

Pandas ima nekoliko tipova za rad sa vremenom:

*Timestamp / DatetimeIndex*

*Timedelta / TimedeltaIndex*

*Period / PeriodIndex*

*DateOffset / None*

# Rad sa vremenom

Kreiranje jednog datuma-vremena

```
dt = pd.Timestamp('2020-01-21 12:00:00')
```

Kreiranje vremenskog opsega

```
rng = pd.date_range('2020-01-01',periods=30,freq="D")
```

Konverzija u datum-vreme

```
dt = pd.to_datetime('2020-01-01 13:25:15')
```

Formatiranje i izvlačenje datuma

```
dt = pd.to_datetime('2020-01-01 13:25:15', format='%Y-%m-%d').date()
```

Formatiranje datetime kolone

```
tbl["time"] = pd.to_datetime(tbl["time"], format='%Y-%m-%d').dt.date
```

# Vežba

(pdap-ex01 players.py)

Izvor: gamestats.txt

Prikazati id-ove 10 igrača sa najviše odigranih partija

| player_id | game |
|---|---|
| '719116' | 6855 |
| '1349252' | 6467 |
| '1310093' | 3724 |
| '1418526' | 3320 |
| '997434' | 2326 |
| '1392853' | 2251 |
| '1426697' | 1718 |
| '702955' | 1588 |
| '1360638' | 1257 |
| '1187201' | 1132 |

# Vežba

(pdap-ex01 mostplayed.py)

Izvor: gamestats.txt

Prikazati kojih se datuma najviše igralo za period od 1 do 15 decembra 2019 godine